**Доверенный искусственный интеллект**

# Объяснимость моделей

# искусственного интеллекта и машинного обучения

А.Е. Самотуга, П.С. Ложников

Конспект лекции

**ОБЪЯСНИМОСТЬ МОДЕЛЕЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И МАШИННОГО ОБУЧЕНИЯ**

Модели глубокого обучения и нейронные сети работают на основе скрытых слоев (т.е. между входными данными и выходным результатом существует несколько уровней математической обработки и принятия решений), но демонстрируют результаты лучше, чем базовые алгоритмы машинного обучения. Использование таких моделей в сферах с повышенным уровнем ответственности за принятые решения привело к созданию концепции объяснимого искусственного интеллекта, в рамках которой исследуются методы анализа или дополнения моделей ИИ, позволяющие сделать внутреннюю логику и выходные данные алгоритмов прозрачными и интерпретируемыми для человека.

Ключевое определение из ГОСТ Р 59276-2020 - объяснимость (explainability). Представляет собой свойство системы искусственного интеллекта, заключающееся в возможности представления причин, приводящих к тому или иному решению системы, в виде, понятном человеку.

## Прозрачность и объяснимость ИИ. Цели обеспечения объяснимости ИИ

По мере расширения возможностей ИИ, людям становится все сложнее осознать, каким образом алгоритм пришел к тому или иному результату. Процесс вычисления превращается в так называемый «черный ящик», который не поддается интерпретации. Модели, работающие по принципу черного ящика, создаются непосредственно на основе данных. При этом даже создавшие алгоритм инженеры и специалисты по обработке и анализу данных не в состоянии понять и объяснить, что именно в них происходит и как алгоритм пришел к конкретному результату.

Объяснимый искусственный интеллект (XAI) — это набор процессов и методов, позволяющих пользователям понять, почему именно алгоритмы машинного обучения пришли к тем или иным результатам или выводам. Объяснимый ИИ применяется для описания модели ИИ, ее ожидаемого влияния и потенциальной предвзятости. Он помогает охарактеризовать точность, достоверность и прозрачность модели, предназначенной для принятия решений с помощью ИИ. Объяснимый ИИ играет важнейшую роль для повышения достоверности и надежности производственных моделей ИИ. Кроме того, объяснимость ИИ помогает организациям с большей ответственностью подходить к разработке ИИ.

Предполагается, что объяснимость ИИ будет включать в себя три составляющие: симулируемость, разложимость, алгоритмическую прозрачность.

Симулируемость означает возможность анализа модели человеком; наиболее важным критерием для симулируемости является сложность модели. Простые, но обширные (со слишком большим количеством правил) системы, основанные на правилах не соответствуют этой характеристике, тогда как одиночная нейронная сеть персептрона попадает в нее.

Разложимость означает способность объяснить каждую из частей модели (входные данные, параметры и выходные данные). Громоздкие функции не соответствуют данному критерию.

Алгоритмическая прозрачность означает способность пользователя понять процесс, которому следует модель ИИ, чтобы произвести любой заданный вывод из ее входных данных. Линейная модель ИИ считается прозрачной, потому что ее поверхность ошибок (математическая интерпретация механизма обучения ИИ) понятна и может быть рассмотрена, что дает пользователю достаточно знаний о том, как модель будет действовать в каждой ситуации, с которой он может столкнуться. В глубоких архитектурах ИИ этого не происходит, поскольку поверхность ошибок может быть непрозрачной и ее нельзя полностью наблюдать, соответственно, решение необходимо аппроксимировать с помощью эвристической оптимизации (например, с помощью стохастического градиентного спуска).

Объяснения алгоритмов работы ИИ могут быть представлены в текстовой или визуальной форме.

Текстовые объяснения представляют собой метод создания символов, отображающих логику алгоритма посредством семантического отображения.

Многие из методов визуализации сопровождаются методами уменьшения размерности для упрощения понимания работы модели человеком. Методы визуализации могут сочетаться с другими методами для улучшения их понимания и считаются наиболее подходящим способом представить сложные взаимодействия между переменными, участвующими в модели.

Отсутствие объяснимости может приводить к ситуациям, когда модель присваивает более высокий вес тем входным переменным, которые объективно не должны иметь такого веса.

Злоумышленники могут добавлять небольшие искажения в изображения, которые не повлияют на решение человека, но могут сбить с толку алгоритм, ранее доказавший свою эффективность на не измененных искусственно изображениях

1. **Объяснимость данных. Оценка вклада признаков в результат анализа**

Часто наборы данных, с которыми приходится работать, содержат большое количество признаков, число которых может достигать нескольких сотен и даже тысяч. При построении модели машинного обучения не всегда понятно, какие из признаков действительно для неё важны (т.е. имеют связь с целевой переменной), а какие являются избыточными (или шумовыми).

Удаление избыточных признаков позволяет лучше понять данные, а также сократить время настройки модели, улучшить её точность и облегчить интерпретируемость. Иногда эта задача и вовсе может быть самой значимой, например, нахождение оптимального набора признаков может помочь расшифровать механизмы, лежащие в основе исследуемой проблемы.

Методы отбора признаков обычно делят на 3 категории: фильтры (filter methods), встроенные методы (embedded methods) и обёртки (wrapper methods). Выбор подходящего метода не всегда очевиден и зависит от задачи и имеющихся данных.

Методы фильтрации применяются до обучения модели и, как правило, имеют низкую стоимость вычислений. К ним можно отнести визуальный анализ (например, удаление признака, у которого только одно значение, или большинство значений пропущено), оценку признаков с помощью какого-нибудь статистического критерия (дисперсии, корреляции, X2 и др.) и экспертную оценку (удаление признаков, которые не подходят по смыслу, или признаков с некорректными значениями).

F-тест оценивает степень линейной зависимости между предикторами и целевой переменной, поэтому он лучше всего подойдёт для линейных моделей.

Взаимная информация показывает насколько чётко определена целевая переменная если известны значения предиктора. Этот тип тестов считается самым удобным в использовании - он хорошо работает "из коробки" и позволяет находить нелинейные зависимости.

Встроенные методы выполняют отбор признаков во время обучения модели, оптимизируя их набор для достижения лучшей точности. К этим методам можно отнести регуляризацию в линейных моделях (обычно L1) и расчёт важности признаков в алгоритмах с деревьями. Отметим, что для линейных моделей требуется масштабирование и нормализация данных.

В зависимости от того, основан анализ на регрессии или классификации, алгоритмы отбора признаков могут отличаться, но главная идея их реализации остается одной и той же. Кроме описанных, для отсбора используются P-значения, Прямой отбор, Обратный отбор, Рекурсивное исключение признаков, Диаграмма важности признаков, Регуляризация.

1. **Объяснимость причинно-следственных связей решений ИИ**

Современный искусственный интеллект не способен принимать решения на основе обнаруженных причинно-следственных связей. Пока что алгоритмы умеют выявлять только закономерности в данных, которые обрабатывают.

Алгоритмы машинного обучения, особенно глубокие нейронные сети, очень хороши в выявлении тонких закономерностей в огромных наборах данных. Они могут преобразовывать звук в реальном времени, размечать тысячи изображений и видеокадров в секунду, а также проверять рентгеновские снимки и МРТ на предмет наличия злокачественных образований. Но им сложно делать простые причинно-следственные выводы, подобные тем, которые мы только что продемонстрировали на примере видео про бейсбол.

В статье под названием «К обучению каузальных представлений» исследователи из Института интеллектуальных систем Макса Планка, Монреальского института алгоритмов обучения (Mila) и Google Research обсуждают проблемы, возникающие из-за отсутствия каузальных представлений в моделях машинного обучения и предоставляют инструкции по созданию систем искусственного интеллекта, которые способные обучать каузальные представления.

Это одна из нескольких попыток, направленных на изучение и решение проблемы отсутствия внятной причинно-следственной методологии в машинном обучении, которая может стать ключом к преодолению некоторых из проблем.

Машинное обучение часто игнорирует информацию, которую живые организмы активно используют: вмешательство в окружающий мир, сдвиги предметной области, временные структуры - в целом, мы считаем эти факторы помехой и пытаемся устранить их. В дополнение к этому, большинство текущих успехов машинного обучения сводятся к крупномасштабному распознаванию образов на надлежащим образом собранных независимых и идентично распределенных данных(iid).

Для генерализации за пределы iid-сетапа требуется изучение не просто статистических ассоциаций между переменными, но и лежащей в основе причинно-следственной (каузальной) модели. Каузальные модели остаются устойчивыми, когда какие-либо вмешательства изменяют статистическое распределение задачи. Причинно-следственная связь также может иметь решающее значение для борьбы с состязательными атаками. В широком смысле причинно-следственная связь может решить проблему отсутствия генерализации в машинном обучении.

Исследователи ИИ объединяют несколько концепций и принципов, которые могут иметь важное значение для создания каузальных моделей машинного обучения.

Две из этих концепций включают «структурные каузальные модели» и «независимые каузальные механизмы». В целом, принципы гласят, что вместо поиска поверхностных статистических корреляций система ИИ должна иметь возможность идентифицировать каузальные переменные и разделять их влияние на среду.

1. **Объяснимость на разных этапах жизненного цикла ИИ**

Объяснимость может быть рассмотрена на всех этапах разработки ИИ, а именно: перед моделированием, в процессе разработки модели и после моделирования.

Самый простой способ добиться интерпретируемости – использовать только подмножество алгоритмов, создающих интерпретируемые модели ИИ. Линейная регрессия, логистическая регрессия и дерево решений являются обычно используемыми интерпретируемыми моделями ИИ. Когда модели машинного обучения не соответствуют ни одному из критериев, налагаемых для объявления их прозрачными, необходимо разработать и применить к модели отдельный метод для объяснения ее решений. Это цель методов апостериорной объяснимости, которые нацелены на передачу понятной информации о том, как уже разработанная модель производит свои прогнозы для любого заданного входа.

Большим преимуществом методов интерпретации, не зависящих от модели, по сравнению с методами интерпретации для конкретных моделей является их гибкость.

Желательными аспектами системы объяснения, не зависящей от модели, являются: гибкость модели (метод интерпретации может работать с любой моделью машинного обучения, такой как случайные леса и глубокие нейронные сети); гибкость объяснений (различные формы объяснений, например, линейная формула для одних случаев, график с важностью функций – для других); гибкость представления (система объяснения должна иметь возможность использовать другое представление объекта в качестве объясняемой модели).

1. **Определение соответствия моделей или данных этическим нормам**

Этические проблемы, связанные с работой моделей черного ящика, возникают из-за их склонности непреднамеренно принимать несправедливые решения с учетом чувствительных факторов, таких как раса, возраст или пол человека. Пример: система оценки риска повторного совершения преступных деяний COMPAS принимала расу как одну из важных характеристик для принятия решения о высоком риске рецидива.

1. **Методы и подходы к реализации объяснимого ИИ и метрики объяснимости (деревья решений, GA2M, TCAV, LIME, SHAP и другое)**

LIME (Local interpretable model-agnostic explanations, локально интерпретируемое не зависящее от модели объяснение) объясняет классификатор для конкретного единичного выхода функции и поэтому подходит для локального рассмотрения. Объясняет отдельные прогнозы нейронной сети, аппроксимируя ее локально с помощью интерпретируемых моделей, таких как линейные модели и мелкие деревья. LIME работает с табличными данными, текстом и изображениями.

SHAP (Аддитивное объяснение Шепли) — метод объяснения индивидуальных прогнозов. SHAP основан на игре теоретически оптимальных значений Шепли. SHAP определяет предельный вклад каждой функции в достижение выходного значения, начиная с базового значения.

Значения Шепли рассматривают все возможные прогнозы с использованием всех возможных комбинаций входных данных. Благодаря такому подходу SHAP может гарантировать согласованность и локальную точность.

SHAP хорошо работает для задач классификации и регрессии, но плохо применим для обучения с подкреплением.

LIME и SHAP были признаны научным сообществом как самые многообещающие модели апостериорных объяснений, не зависящих от модели, однако группой ученых было проведено исследование, доказавшее, что данные методы плохо определяют необъективность модели. В эксперименте со специально созданным классификатором, который был явно необъективен, используя в качестве значимых признаков только данные о расе человека, данные методы игнорировали необъективность, находя вполне безобидные объяснения для полученных в ходе работы модели выходных данных.

1. **Программные продукты для создания объяснимого ИИ или повышения объяснимости ИИ**

AIX360 от IBM — расширяемый набор инструментов, который предлагает ряд возможностей для улучшения объяснимости модели. Соответствующий пакет Python AI Explainability 360 включает в себя алгоритмы, охватывающие различные метрики объяснений наряду с посредническими метриками объяснимости, предоставляет инструменты для визуального исследования поведения обученных моделей с помощью минимального количества кода.

Интерпретируемость модели в Microsoft Azure обеспечивается пакетом SDK, используя который можно объяснять предсказания модели через генерацию важности значения функций для всей модели или отдельных точек данных и с помощью интерактивной визуализации для обнаружения связей в данных и объяснений во время обучения модели.

Azureml-interpret использует методы интерпретируемости, разработанные в Interpret-Community, пакете Python с открытым исходным кодом для обучения интерпретируемых моделей и помощи в объяснении систем искусственного интеллекта с «черным ящиком». Interpret-Community служит базой для поддерживаемых моделей объяснения этого SDK и в настоящее время поддерживает следующие методы интерпретируемости: вариации SHAP (TreeSHAP, SHAP deep Explainer, SHAP Kernel explainer и др.), Global Surrogate1, Permutation Feature Importance Explainer2. Для объяснения работы глубоких нейронных сетей можно использовать TabularExplainer, который использует методы SHAP.

Google Cloud’s AI Explanations ставит целью использовать объяснения ИИ для упрощения разработки модели, а также объяснить поведение модели ключевым заинтересованным сторонам. AI Explanations работает с моделями, решающими задачи классификации и регрессии, демонстрируя, как та или иная функция данных повлияла на результат. В AI Explanations используются следующие методы: метод интегрированных градиентов, XRAI и Sampled Shapley. Для визуализаций используется What-If Tool.

В 2018 году Команда PAIR (People + AI Reserach, часть Google AI) представила What-If Tool — инструмент для обнаружения предвзятости в моделях искусственного интеллекта. Он поставляется как часть веб-приложения TensorBoard. What-If Tool визуализирует влияние определенных данных на предсказание модели, при этом доступен для тех, кто не разбирается в программировании.

Платформа Thales XAI обеспечивает различные уровни объяснения, например, на основе примеров, на основе функций, контрфактов с использованием текстовых и визуальных представлений, однако стоит отметить, что упор делается на объяснение на основе семантики с помощью графов знаний. Графы знаний используются для кодирования лучшего представления данных, структурирования модели машинного обучения более интерпретируемым образом и принятия семантического сходства для локального (на основе экземпляров) и глобального (на основе модели) объяснения.

Объяснение любого предсказания связи получается путем определения репрезентативных горячих точек в графе знаний, то есть связанных частей графиков, которые при удалении отрицательно влияют на точность прогнозирования.

ELI5 – это пакет Python, который помогает отлаживать классификаторы машинного обучения и объяснять их прогнозы. Он обеспечивает поддержку нескольких платформ и пакетов машинного обучения.

ELI5 также реализует несколько алгоритмов проверки моделей черного ящика:

TextExplainer позволяет объяснять предсказания любого текстового классификатора с использованием алгоритма LIME. Существуют также утилиты для использования LIME с нетекстовыми данными и произвольными классификаторами черного ящика, но эта функция в настоящее время является экспериментальной.

Некоторые программные пакеты XAI, доступные в GitHub:

Interpret от InterpretML может использоваться для объяснения моделей черного ящика и в настоящее время поддерживает объяснимый бустинг, деревья решений, список правил принятия решений, линейную логистическую регрессию, SHAP kernel explainer, TreeSHAP, LIME, анализ чувствительности Морриса и частичную зависимость.

Пакет IML охватывает такие методы как важность функций, графики частичной зависимости, графики индивидуальных условных ожиданий, накопленные локальные эффекты, суррогаты деревьев, LIME и SHAP.

Пакет DeepExplain включает различные методы на основе градиента, такие как карты значимости, интегрированные градиенты, DeepLIFT, LRP и др., а также методы на основе возмущений, такие как окклюзия, SHAP и др.

**Список источников**

1. Корешкова Т. Объяснимый искусственный интеллект [Электронный ресурс]. URL:<https://rdc.grfc.ru/2020/12/explainable-ai/> (дата обращения: 12.04.2022).

2. Отбор признаков в задачах машинного обучения. Часть 1 [Электронный ресурс]. URL:<https://habr.com/ru/post/550978/> (дата обращения: 12.04.2022).

3. ISO/IEC AWI TS 6254 [Электронный ресурс]. URL:<https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/08/21/82148.html> (дата обращения: 13.04.2022).